Divide and Conquer: Problem 1

We have a vector V [1..N] formed by integers, so that all of them are different, and that they are ordered in an increasing way. It is said that a vector of these characteristics is coincident if it has at least one position such that it is equal to the value that contains the vector in that position. For example, in the vector

1	2	3	4	5	6	7	8
-14	-6	3	6	16	18	27	43

it can be seen that V [3] = 3; therefore, this vector is coincident.

Design a Divide and Conquer algorithm that determines in an order of efficiency not greater than O(logn) if a vector is coincident, receiving as data the vector and its size.

Divide and Conquer: Problem 1

```
const MaxSize= ...
type vector = array[1.. MaxSize] of integer
fun IsCoincident(I/O xi, xf: integer; I/O v: vector)  ret e: boolean
var m: integer
    e ← false                                                        (1)
    if ( xi == xf ) then                 //Basic case                (1)
        if ( v(xi) == xi) then e ← true eif                          (1+1)
    else
        m = ( xi + xf ) % 2                                          (1)
        if ( v(m) == m) then e ← true                                (1+1)
        else
            if ( v(m) < m) then                    //Upper half      (1)
                xi = m                                               (1)
            else                                   //Lower half
                xf = m                                               (1)
            eif
            e = IsCoincident (xi, xf, v)                             T(n/2)
        eif
    eif
    return e                                                         (1)
ffun
```

$$T(n) = 7 + T\left(\frac{n}{2}\right)$$

Divide and Conquer: Problem 1

```
const MaxSize= ...
type vector = array[1.. MaxSize] of integer
fun IsCoincident(I/O xi, xf: integer; I/O v: vector)  ret e: boolean
var m: integer
     e ← false                                                   (1)
     if ( xi == xf ) then           //Basic case                 (1)
          if ( v(xi) == xi) then e ← true eif                    (1+1)
     else
          m = ( xi + xf ) % 2                                    (1)
          if ( v(m) == m) then e ← true                          (1+1)
          else
               if ( v(m) < m) then          //Upper half         (1)
                    xi = m+1                                          (1)
               else                         //Lower half
                    xf = m-1                                          (1)
               eif
               e = IsCoincident (xi, xf, v)                      T(n/2)
          eif
     eif
     return e                                                    (1)
ffun
```

$$T(n) = 7 + T\left(\frac{n}{2}\right)$$

Divide and Conquer: Problem 1

$$T(n) = 7 + T\left(\frac{n}{2}\right)$$

- Step 2: Polynomical equation

$$n = 2^k \Leftrightarrow k = \log_2 n \quad \rightarrow \quad x^k - x^{k-1} = 7$$

- Step 3: Solution of the homogeneous part

$$x^{k-1} \cdot (x-1) = 0 \quad \rightarrow \quad \text{Raíz: } (x-1)^1$$

- Step 4: Solution of the non homogeneous part

$$7 = 7k^{0} \cdot 1^k \quad \rightarrow \quad \text{Raíz: } (x-1)^{0+1}$$

- Step 5: Combination of roots

$$(x-1)^2$$

$$T(k) = c_1 + c_2 \cdot k \rightarrow T(n) = c_1 + c_2 \cdot \log n$$

$$\boldsymbol{T(n)} \in \boldsymbol{O}(\log n)$$

Divide and Conquer: Problem 1

$$T(n) = 7 + T\left(\frac{n}{2}\right)$$

- Is a typical Divide and Conquer expression qith p=0, b=2 and k=1. Then:

$$\mathrm{k} = b^p$$

- Therefore:

$$\boldsymbol{T(n)} \in \boldsymbol{O}(n^p \log n) \in \boldsymbol{O}(\log n)$$